NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

SOME COMPUTER ALGORITHMS TO IMPLEMENT
A RELIABILITY SHORTHAND

Sadan Gursel

October 1982

Thesis Advisor: J. D. Esary

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	
2. GOVT ACCESSION NO.	
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle)	Some Computer Algorithms to Implement a Reliability Shorthand
5. TYPE OF REPORT & PERIOD COVERED	Master's Thesis October 1982
6. PERFORMING ORG. REPORT NUMBER	
7. AUTHOR(s)	Sadan Gursel
8. CONTRACT OR GRANT NUMBER(s)	
9. PERFORMING ORGANIZATION NAME AND ADDRESS	Naval Postgraduate School Monterey, California 93940
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS	
11. CONTROLLING OFFICE NAME AND ADDRESS	Naval Postgraduate School Monterey, California 93940
12. REPORT DATE	October 1982
13. NUMBER OF PAGES	59
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	
15. SECURITY CLASS. (of this report)	Unclassified
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release, distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

reliability
survival function
reliability shorthand
system survival
systems
redundant systems

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Under the assumption of constant failure rates it is possible to build a "reliability shorthand" which gives a simple, unified approach to reliability computations for systems in the presence of complications like support by shared spares or changes in the failure rates of surviving components when other components fail. The computational implementation of the shorthand depends upon the convolution of strings of exponentially distributed random

variables.

This paper presents an algorithm for the numerical convolution of exponentially distributed random variables. After reducing the system scenario to its shorthand format, one can use the programs that are given in the appendix to obtain numerical values for the reliability of the system.

Some Computer Algorithms to Implement a Reliability Shorthand

by

Sadan Gursel
Lieutenant Junior Grade, Turkish Navy

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
October, 1982

ABSTRACT

Under the assumption of constant failure rates it is possible to build a "reliability shorthand" which gives a simple, unified approach to reliability computations for systems in the presence of complications like support by shared spares or changes in the failure rates of surviving components when other components fail. The computational implementation of the shorthand depends upon the convolution of strings of exponentially distributed random variables.

This paper presents an algorithm for the numerical convolution of exponentially distributed random variables. After reducing the system scenario to its shorthand format, one can use the programs that are given in the appendix to obtain numerical values for the reliability of the system.

TABLE OF CONTENTS

I. INTRODUCTION

The reliability shorthand considered in this paper has been developed in conjunction with the course OA 4302 "Reliability And Weapon Systems Effectiveness Measurement" at the Naval Postgraduate School. A tutorial introduction to the reliability shorthand was given by Repicky[Ref.2]. This paper is devoted to a complementary part of the idea.

Any study on system reliability always requires two steps; one is the description of the system's life and the other is the derivation of its survival function.

Under the assumption of constant component failure rates this paper presents a way of obtaining the system's reliability which requires little beyond the description of the system's life.

Section II contains an approach to the convolution of exponentially distributed random variables. Also there is a presentation of a computational algorithm for the convolution of exponentially distributed random variables.

Appendix A gives survival functions corresponding to several reliability shorthand notations and a program writen in Fortran for computations from shorthand notations.

Section III deals with the reliability of redundant systems under the assumption of constant component failure rates.

Appendix B consists of a program written in Fortran. The program supports the approach of Section III. There is a crude Monte Carlo simulation program in Appendix D which is a simulation program parallel to the program in Appendix B.

There is another program in Appendix C written in Fortran. This program uses the network approach to systems described in Section III.

Appendix E summarizes the definitions used in this paper.

II. AN APPROACH TO COMPUTING CONVOLUTIONS OF EXPONENTIAL RANDOM VARIABLES

This section introduces a general algorithm for computing the survival function of any convolution of exponential random variables.

In reliability, the term convolution refers to the summation of independent random lives. In order to have simplicity in specifying convolutions, the reliability shorthand introduces a special notation. In the following sections we will use this notation.

A. THE SURVIVAL FUNCTION FOR A CONVOLUTION OF RANDOM VARIABLES

Let $\overline{F}1(t)$ and $\overline{F}2(t)$ be the survival functions for the random variables T1 and T2 respectively. Let f1(t) and f2(t) be the corresponding densities. Let $\overline{F}(t)$ be the survival function for the random variable T, where T=T1+T2.

Then the likelihood expression for $\overline{F}(t)$ is

$$\overline{F}(t)=\overline{F}1(t)+\int_0^t \overline{F}2(t-s)\ f1(s)\ ds$$

In the right hand side of equation, $\overline{F}1(t)$ is the probability that component one completes the mission, $\int_0^t \overline{F}2(t-s)f1(s)ds$ is the probability that at some time s $(0 \le s \le t)$ component one fails, component two takes its place and carries the system to the end of the mission duration t.

In order to illustrate consider some applications.

1. Example

Reliability Shorthand Notation : $EXP\{\lambda_1\}+EXP\{\lambda_2\}$

SYSTEM: One component having one spare with a dissimilar failure rate. If the active component fails, the spare will replace it immediately.

Here the life for the system is T = T1 + T2. The reliability shorthand notation indicates that this system has an exponential life with failure rate λ_1 followed by an exponential life with failure rate λ_2.

The survival function for the active component is

$$\overline{F}_1(t) = e^{-\lambda_1 t},\ t \ge 0.$$

The survival function for the spare is

$$\overline{F}_2(t) = e^{-\lambda_2 t},\ t \ge 0.$$

The survival function for the system is

$$\bar{F}(t) = \bar{F}_1(t) + \int_0^t \bar{F}_2(t-s)\, f_1(s)\, ds$$

$$\bar{F}(t) = e^{-\lambda_1 t} + \int_0^t e^{-\lambda_2(t-s)}\, \lambda_1 e^{-\lambda_1 s}\, ds \quad , \; t \geq 0.$$

If we complete the integration, the result is

$$\bar{F}(t) = \lambda_2/(\lambda_2 - \lambda_1)\, e^{-\lambda_1 t} + \lambda_1/(\lambda_1 - \lambda_2)\, e^{-\lambda_2 t} \quad , \; t \geq 0.$$

Which is the well known result.

Another way to establish this formula is the use of the moment generating function. (Freund and Walpole[Ref.3])

2. Example

Reliability Shorthand Notation : EXP{ λ } + EXP{ λ }

SYSTEM: One component having one identical spare. If the active component fails, the spare will replace it immediately.

The formula that we derived above becomes meaningless, because the denominators become zero. If we proceed as before

$$\bar{F}(t) = \bar{F}1(t) + \int_0^t \bar{F}2(t-s)\; f1(s)\; ds$$

$$\bar{F}(t) = e^{-\lambda t} + \int_0^t e^{-\lambda(t-s)} \lambda e^{-\lambda s}\, ds$$

$$\bar{F}(t) = (1 + \lambda t)\, e^{-\lambda t} \quad , \; t \geq 0.$$

The result comes out as expected to be the Erlang $\{2,\lambda\}$ survival function.

B. RELIABILITY SHORTHAND NOTATION

EXP$\{\lambda\}$+EXP$\{\lambda\}$+....+EXP$\{\lambda\}$ (n identical exponential lives)

This leads to the Erlang $\{n,\lambda\}$ survival function

$$\bar{F}(t) = \sum_{i=1}^{n} (\lambda t) \;/(i-1)! \quad e^{-\lambda t}, \; t \geq 0.$$

The use of moment generating function gives the result immediately.

C. THE RELIABILITY SHORTHAND NOTATION

EXP$\{\lambda_1\}$+EXP$\{\lambda_2\}$+............+EXP$\{\lambda_n\}$

This is the expression for the convolution of n random variables where each has a distinct failure rate.

By the approach used in Section 2.1, adding one exponential life at once, one can obtain the formula for the survival function

$$\bar{F}(t) = \sum_{i=1}^{n} \prod_{j\neq i} \lambda_j \;/ \prod_{j\neq i} (\lambda_1 - \lambda_2) \quad e^{-\lambda_i t} \;, \; t \geq 0.$$

This is also a well known formula and can be obtained from by a moment generating function.

D. THE RELIABILITY SHORTHAND NOTATION

$$\mathrm{EXP}\{\lambda_1\}+\mathrm{EXP}\{\lambda_2\}+\ldots\ldots\ldots\ldots+\mathrm{EXP}\{\lambda_2\} \qquad (n_1 \text{ terms})$$

$$+\ \mathrm{EXP}\{\lambda_2\}+\mathrm{EXP}\{\lambda_2\}+\ldots\ldots\ldots\ldots+\mathrm{EXP}\{\lambda_2\} \qquad (n_2 \text{ terms})$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$+\ \mathrm{EXP}\{\lambda_k\}+\mathrm{EXP}\{\lambda_k\}+\ldots\ldots\ldots\ldots+\mathrm{EXP}\{\lambda_k\} \qquad (n_k \text{ terms})$$

This is the convolution of $\sum_{i=1}^{k} n_i$ exponential random variables where there are n_i identical exponential random variables having the failure rate λ_i.

The moment generating function technique is not useful in this situation, since there is a huge amount of complexity involved. This section deals with this notation using the convolution formula.

1. Reliability Shorthand Notation

$$\mathrm{EXP}\{\lambda_1\}+\mathrm{EXP}\{\lambda_1\}+\mathrm{EXP}\{\lambda_2\}$$

We know the survival function for $\mathrm{EXP}\{\lambda_1\}+\mathrm{EXP}\{\lambda_2\}$ and also for $\mathrm{EXP}\{\lambda\}+\mathrm{EXP}\{\lambda\}$. The next two subsections present different ways to reach the survival function for the shorthand notation above.

(a).

Let T1,T2,T3 be random variables distributed as EXP$\{\lambda_1\}$, EXP$\{\lambda_1\}$, EXP$\{\lambda_2\}$ respectively.

Let T1'=T1+T2. This random variable has the Erlang $\{2,\lambda\}$ distribution.

The convolution formula for T=T1'+T3 is;

$$\bar{F}_T(t)=\bar{F}_{T_1'}(t) + \int_0^t \bar{F}_{T_3}(t-s)\ f_{T_1}(s)\ ds \quad , t\geq 0.$$

$$\bar{F}_T(t)=(1+\lambda_1 t)e^{-\lambda_1 t}+\int_0^t e^{-\lambda_2(t-s)}\lambda_1^2 s\ e^{-\lambda_1 s}ds$$

$$\bar{F}_T(t)=e^{-\lambda_2 t}\{(\lambda_2^2-2\lambda_1\lambda_2)/(\lambda_2-\lambda_1)^2 \quad + \lambda_1\lambda_2/(\lambda_2-\lambda_1)\ t\}$$
$$+e^{-\lambda_2 t}\ \{\ \lambda_1^2/(\lambda_1-\lambda_2)^2\ \}\ ,t\geq 0.$$

(b).

Let T1,T2,T3 be random variables distributed as EXP$\{\lambda_1\}$, EXP$\{\lambda_1\}$, EXP$\{\lambda_2\}$ respectively.

Let T2'=T2+T3. This random variable has the survival function

$$\bar{F}_{T_2}(t) = \lambda_2/(\lambda_2-\lambda_1)e^{-\lambda_1 t} + \lambda_1/(\lambda_1-\lambda_2)e^{-\lambda_2 t} \quad ,t\geq 0.$$

The convolution formula for T=T1+T2' is

$$\bar{F}_T(t) = \bar{F}_{T_2'}(t) + \int_0^t \bar{F}_{T_1}(t-s)\ f_{T_2'}(s)\ ds \quad , t\geq 0.$$

$$\bar{F}_T(t) = \lambda_2/(\lambda_2-\lambda_1)\, e^{-\lambda_1 t} + \lambda_1/(\lambda_1-\lambda_2)\, e^{-\lambda_2 t} + \int_0^t e^{-\lambda_1(t-s)} \lambda_1\lambda_2/(\lambda_2-\lambda_1)\,(e^{-\lambda_1 s} - e^{-\lambda_2 s})\, ds$$

$$\bar{F}_T(t) = e^{-\lambda_1 t}\{(\lambda_2^2 - 2\lambda_1\lambda_2)/(\lambda_2-\lambda_1)^2 + \lambda_1/(\lambda_2-\lambda_1)\, t\} + e^{-\lambda_2 t}\{\lambda_1^2/(\lambda_1-\lambda_2)^2\}, t \geq 0.$$

Subsections (a) and (b) illustrate that the convolution formula gives a unique result, regardless of the way of choosing the prior random life.

2. Reliability Shorthand Notation

$EXP\{\lambda_1\} + EXP\{\lambda_1\} + EXP\{\lambda_2\} + EXP\{\lambda_2\}$

Let T1, T2, T3, T4 be random variables distributed as

$EXP\{\lambda_1\}$, $EXP\{\lambda_1\}$, $EXP\{\lambda_2\}$, $EXP\{\lambda_2\}$ respectively.

Then from the derivation in Section 2, T1'=T1+T2+T3 is a random variable having the survival function

$$\bar{F}_{T_1'}(t) = (a_{11} + a_{12} t) e^{-\lambda_1 t} + a_{21} e^{-\lambda_1 t}, t \geq 0.$$

where $a_{11} = (\lambda_2^2 - 2\lambda_1\lambda_2)/(\lambda_2-\lambda_1)^2$, $a_{12} = \lambda_1\lambda_2/(\lambda_2-\lambda_1)$,

$a_{21} = \lambda_1^2/(\lambda_1-\lambda_2)^2$.

The convolution formula for T=T1'+T4 is

$$\bar{F}_T(t) = \bar{F}_{T_1'}(t) + \int_0^t \bar{F}_{T_4}(t-s) f_{T_1'}(s)\, ds , \ t \geq 0.$$

$$\bar{F}_T(t) = (a_{11} + a_{12} t) e^{-\lambda_1 t} + a_{21} e^{-\lambda_2 t} + \int_0^t e^{-\lambda_2(t-s)} ((a_{11}\lambda_1 - a_{12} + \lambda_1 a_{12} s) e^{-\lambda_1 s} + \lambda_2 a_{21} e^{-\lambda_2 s})\, ds$$

The result from the above is

$$\bar{F}_T(t) = (a'_{11} + a'_{12}t)\, e^{-\lambda_1 t} + (a'_{21} + a'_{22}t)\, e^{-\lambda_1 t}, t \ge 0.$$

where $a'_{11} = (\lambda_2^3 - 3\lambda_1^2\lambda_1)/(\lambda_2 - \lambda_1)^3$, $a'_{12} = \lambda_1\lambda_2^2/(\lambda_2 - \lambda_1)^2$,

$a'_{21} = (\lambda_1^3 - 3\lambda_1^2\lambda_2)/(\lambda_1 - \lambda_2)^3$, $a'_{22} = \lambda_1^2\lambda_2/(\lambda_1 - \lambda_2)^2$.

It is important to note that the number of exponential terms equals the number of dissimilar failure rates and each exponential term has a polynomial coefficient with the degree of the polynomial equal to the number of identical random variables having the corresponding failure rate.

The next section deals with the convolution of exponentially distributed random variables using the fact illustrated above.

3. Introduction of an Algorithm for the Convolution of Exponential Lives

From Subsection 2, we can infer the form of the survival function which we sought at the beginning of Section C.

SHORTHAND NOTATION :

$\text{EXP}\{\lambda_1\} + \text{EXP}\{\lambda_1\} + \ldots\ldots + \text{EXP}\{\lambda_1\}$ (n_1 terms)

$+ \text{EXP}\{\lambda_2\} + \text{EXP}\{\lambda_2\} + \ldots\ldots + \text{EXP}\{\lambda_2\}$ (n_2 terms)

....

$+ \text{EXP}\{\lambda_k\} + \text{EXP}\{\lambda_k\} + \ldots\ldots + \text{EXP}\{\lambda_k\}$ (n_k terms)

SURVIVAL FUNCTION :

$$\bar{F}(t) = A_1(t)e^{-\lambda_2 t} + A_2(t)\, e^{-\lambda_2 t} + \ldots + A\ (t)\, e^{-\lambda_k t}\ ,\ t \geq 0$$

where $A_1(t) = a_{11} + a_{12}t + a_{13}t^2 + \ldots + a_{2n_1} t^{n_1 - 1}$,

$A_2(t) = a_{21} + a_{22}t + a_{23}t^2 + \ldots + a_{2n_2} t^{n_2 - 1}$,

....

$A_k(t) = a_{k1} + a_{k2}t + a_{t3}t^2 + \ldots + a_{kn_k} t^{n_k - 1}$.

a. Example:

SHORTHAND NOTATION :

$EXP\{\lambda_1\} + EXP\{\lambda_2\}$

$+\ EXP\{\lambda_2\}$

$+\ EXP\{\lambda_3\} + EXP\{\lambda_3\} + EXP\{\lambda_3\}$

SURVIVAL FUNCTION :

$$\bar{F}(t) = (a_{11} + a_{12}t)e^{-\lambda_1 t} + a_{21}\, e^{-\lambda_2 t} + (a_{31} + a_{32}t + a_{33}t^2)\, e^{-\lambda_2 t}, t \geq 0.$$

b. An Algorithm to Compute the Coefficients

The algorithm represented below develops the survival function by adding one random variable in each run. As an example , in order to compute the survival function for the convolution of ten exponentially distributed random variables, the algorithm is supposed to run ten times.

The notation used in the algorithm is:

K number of dissimilar failure rates

λ_i ith type failure rate

λ_{i_e} failure rate for the currently entering life

a_{jk} kth coefficient on the jth polynomial

n_i current number of identical lives having the ith failure rate

nn_i number of random variables having ith failure rate.

Initial: $a_{jk}=0$, $\forall$ j,k where $j=1,2,..K$, $k=1,2,...n_j$

$n_i=0$, $\forall i$ where $i=1,2,..K$

Input : λ_i , $\forall i$ where $i=1,2,...K$

nn_i, $\forall$ i where $i=1,2,...K$

The first run is : $n_1=1$, $a_{11}=1$

Algorithm :

$n_{i_e} = n_{i_e} + 1$, until $n_{i_e} = nn_{i_e}$.

1. Update the coefficients : $a_{i_e k}$ for $k=2,3,..,n_{i_e}$

$a_{i_e k} = \lambda_{i_e} a_{i_e m-1} / (m-1)$, where $m = n_{i_e} - j$ for $j=0,1,.,n_{i_e}-1$

2. Update the coefficient : $a_{i_e 1}$

$$a_{i_e} = a_{i_e 1} + \sum_{\substack{i \neq i_e \\ n_i > 0}} \{ a_{i1} \lambda_1 / (\lambda_2 - \lambda_{i_e}) + \sum_{\substack{j=2 \\ n_i > 1}}^{n_i} (j-1)! \lambda_{i_e} a_{ij} / (\lambda_2 - \lambda_{i_e})^J \}$$

3. Update the other coefficients: a_{ik} $\forall i,k$

where $i \neq i_e$, $n_i \neq 0$ for $i=1,2,.,K$, $k=1,2,.,n_i$

$a_{in_i} = a_{in_i} \lambda_{i_e} / (\lambda_{i_e} - \lambda_i)$, $\forall i$ where $i \neq i$.

$a_{in} = (\lambda_{i_e} a_{in} - m a_{imH}) / (\lambda_{i_e} - \lambda_i)$, $\forall i$ where $i \neq i_e$ and $n_i \geq 1$

for $m = n_i - j$, $j=1,2,..,n_i - 1$

c. Example:

Reliability Shorthand Notation :

$EXP\{\lambda_1\} + EXP\{\lambda_1\} + EXP\{\lambda_2\} + EXP\{\lambda_2\}$

Let T1,T2,T3,T4 be random variables distributed as $EXP\{\lambda_1\}$, $EXP\{\lambda_1\}$, $EXP\{\lambda_2\}$, $EXP\{\lambda_2\}$ respectively.

We would like to derive a formula for the survival function of the random variable T=T1+T2+T3+T4.

The use of algorithm:

1^{st} RUN;

$n_1=1,\ a_{11}=1$

At the end of this run we have only one random variable, which is distributed EXP{ λ_1 }.

2^{nd} RUN;

$i_e=1,\ n_1=2$

$a_{12}=\lambda_1 a_{11}$ then $a_{12}=\lambda_1$

$a_{11}=a_{11}+0$ then $a_{11}=1$

At the end of the 2. run we have the survival function for the T'=T1+T2, where T1 and T2 are identically distributed as EXP{ λ_1 }. The survival function is

$$\bar{F}(t)=(a_{11}+a_{12}t)\,e^{-\lambda_1 t},$$

$$\bar{F}(t)=(1+\lambda_1 t)\,e^{-\lambda_1 t},\quad t\geq 0$$

Which is ERLANG{2, λ_1}.

3^{rd} RUN;

$i_e=2,\ n_1=2,\ n_2=1$

$a_{21} = 0 + a_{11}\lambda_1/(\lambda_1 - \lambda_2) + a\ \ 1!\ \lambda_2/(\lambda_1 - \lambda_2)^2$

$= 1\lambda_1/(\lambda_1 - \lambda_2) + \lambda_1\lambda_2/(\lambda_1 - \lambda_2)^2$

$= \lambda_1^2/(\lambda_1 - \lambda_2)^2$

$a_{12} = a_{12}\lambda_2/(\lambda_2 - \lambda_1)$ then $a_{12} = \lambda_1\lambda_2/(\lambda_2 - \lambda_1)$

$a_{11} = (\lambda_2 a_{11} - a_{12})/(\lambda_2 - \lambda_1)$ then $a_{11} = (\lambda_2^3 - 2\lambda_1\lambda_2)/(\lambda_2 - \lambda_1)^2$

Note that, here the coefficient a_{12} is the updated one.

At the end of the 3. run we have the survival function for the random variable T"=T1+T2+T3, where T1,T2,T3 are as defined before.

$\bar{F}_{T''}(t) = (a_{11} + a_{12}t)\, e^{-\lambda_1 t} + a_{21} e^{-\lambda_2 t},\ t \geq 0$

where $a_{11} = (\lambda_2^3 - 2\lambda_1\lambda_2)/(\lambda_2 - \lambda_1)^2$, $a_{12} = \lambda_1\lambda_2/(\lambda_2 - \lambda_1)$,

$a_{21} = \lambda_1^2/(\lambda_1 - \lambda_2)^2$.

Note that the coefficients are identical to the result of Subsection (b).

4^{th} RUN;

$i_e = 2,\ n_1 = 2,\ n_2 = 2$

$a_{22} = \lambda_2 a_{21}$ then $a_{22} = \lambda_1^2\lambda_2/(\lambda_1 - \lambda_2)^2$

$a_{21} = a_{21} + a_{11}\lambda_1/(\lambda_1 - \lambda_2) + a_{12}\lambda_2/(\lambda_1 - \lambda_2)^2$

then $a_{21} = (\lambda_1^3 - 3\lambda_2^2\lambda_2)/(\lambda_1 - \lambda_2)^3$

$$a_{12} = a_{12}\lambda_2/(\lambda_2 - \lambda_1) \quad \text{then} \quad a_{12} = \lambda_1\lambda_2^2/(\lambda_2-\lambda_1)^2$$

$$a_{11} = (\lambda_2 a_{11} - a_{12})/(\lambda_2 - \lambda_1) \quad \text{then} \quad a_{11} = (\lambda_2^3 - 3\lambda_1\lambda_2^2)/(\lambda_2-\lambda_1)^3$$

At the end of the 4. run, we have the survival function for the variable T=T1+T2+T3+T4.

$$\bar{F}_T(t) = (a_{11}+a_{12}t)\,e^{-\lambda_1 t} + (a_{21}+a_{22}t)\,e^{-\lambda_2 t} \quad ,t\geq 0$$

where $a_{11} = (\lambda_2^3 - 3\lambda_1\lambda_2^2)/(\lambda_2-\lambda_1)^3$, $a_{12} = \lambda_1\lambda_2^2/(\lambda_2-\lambda_1)^2$,

$$a_{21} = (\lambda_1^3 - 3\lambda_1^2\lambda_2)/(\lambda_1-\lambda_2)^3 \ , \quad a_{22} = \lambda_1^2\lambda_2/(\lambda_1-\lambda_2)^2 \ .$$

Note that the coefficients are identical to the result of Subsection 2.

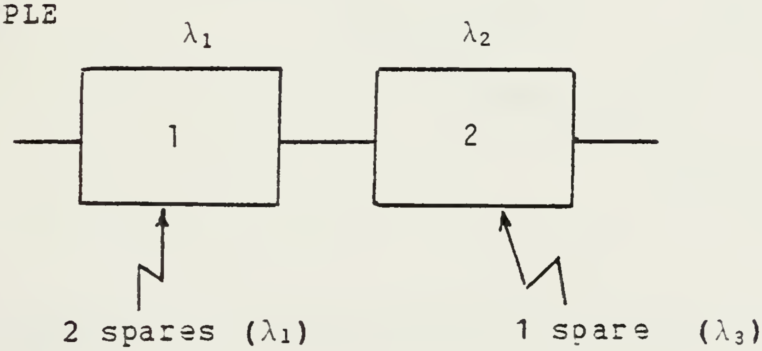
III. RELIABILITY SHORTHAND APPROACH TO SYSTEM RELIABILITY

This section deals with a system whose components have constant failure rates.

Having the reliability network for a system in which each component has an exponential life and knowing the probabilities for failures (discussed in Appendix E2) makes it easy to describe the system's life.

In order to make the idea clear, we will go through some examples.

A. EXAMPLE

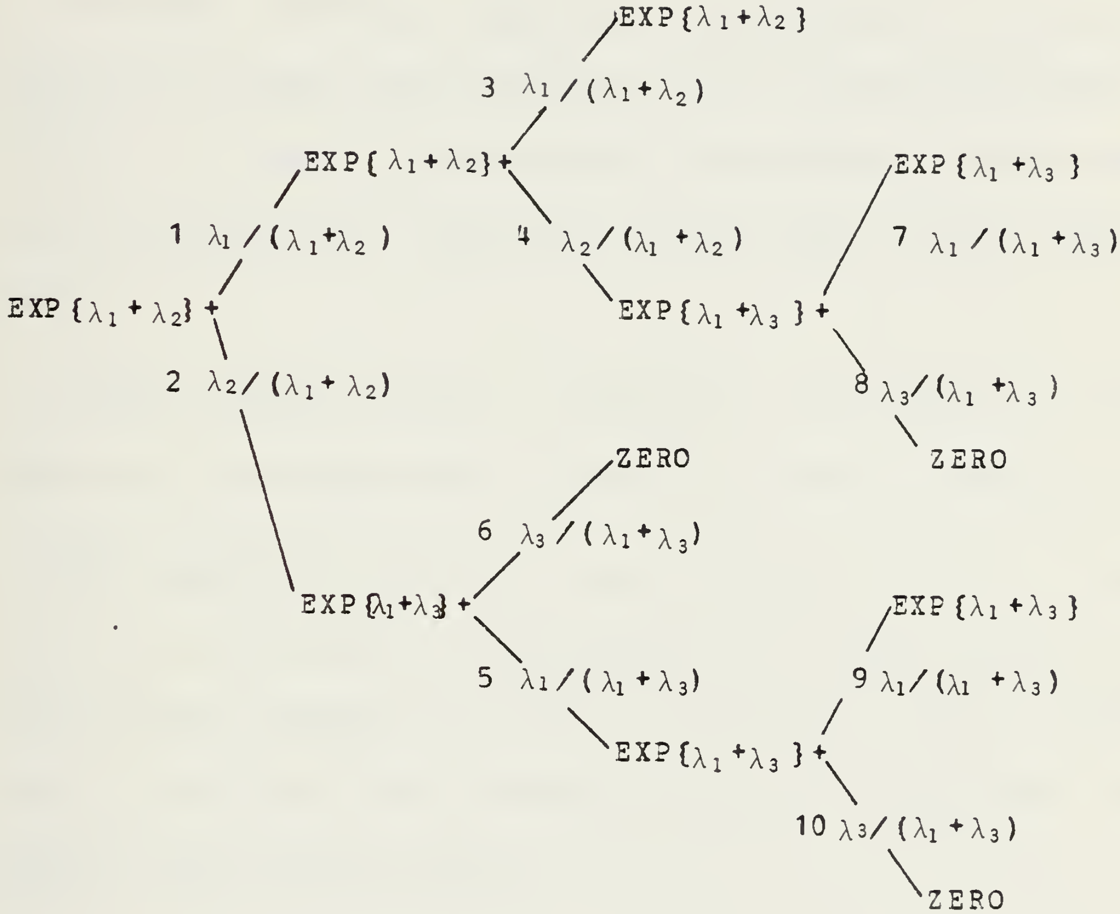


SYSTEM : 2 components in series with spares.

DESCRIPTION : System has 2 components in series. Component 1 has a life distributed as EXP{λ_1} and there are 2 identical spares. Component 2 has a life distributed as

EXP$\{\lambda_2\}$ and there is one non-identical spare, whose life is distributed as EXP$\{\lambda_3\}$.

LIFE :



Explanation for the derivation of system's life:

At the beginning the system has an EXP$\{\lambda_1+\lambda_2\}$ life. The failure of the system is by the failure of component 1 with a probability of $\lambda_1/(\lambda_1+\lambda_2)$ or by the failure of component 2 with probability of $\lambda_2/(\lambda_1+\lambda_2)$.

In the life figure, number 1 denotes the event "failure of component 1" and number 2 denotes the event "failure of component 2".

If event 1 occurs, component 1 is replaced by one of the spares and system again functions with a life distributed as EXP$\{\lambda_1+\lambda_2\}$, since the exponential distribution has the memoryless property and component 2 still has the same failure rate.

If the event 2 occurs, component 2 is replaced by its spare and the system has a life distributed as EXP$\{\lambda_1+\lambda_3\}$.

The numbers on the life figure correspond to the transitions that can occur. The probability on each arc shows the conditional probability of the transition. As an example event 3 can occur with probability of $\lambda_1/(\lambda_1+\lambda_2)$, given that event 1 has occured before.

The distribution ZERO defined by Esary [Ref.1] and Repicky [Ref.2] (also defined in Appendix B3) enters when life is exhausted.

For convenience of description, it is helpful to define the concept of path used in this paper. Path denotes the

sequence of events in the system's life from the starting point to the point where the system is not functioning.

Examples;

Events 1 and 3 are a path, which denotes a sequence of lives for the system. In this case, the system has 3 exponentially distributed lives $EXP\{\lambda_1+\lambda_2\}+EXP\{\lambda_1+\lambda_2\}+EXP\{\lambda_1+\lambda_2\}$ with the probability of $\lambda_1/(\lambda_1+\lambda_2)\ \lambda_1/(\lambda_1+\lambda_2)$.

Events 1, 4 and 8 form another path, which describes a sequence of lives for the system. In this path, the system life is $EXP\{\lambda_1+\lambda_2\}+EXP\{\lambda_1+\lambda_2\}+EXP\{\lambda_1+\lambda_3\}+ZERO$. The probability of this path is

$$[\lambda_1/(\lambda_1+\lambda_2)][\lambda_2/(\lambda_1+\lambda_2)][\lambda_3/(\lambda_1+\lambda_3)].$$

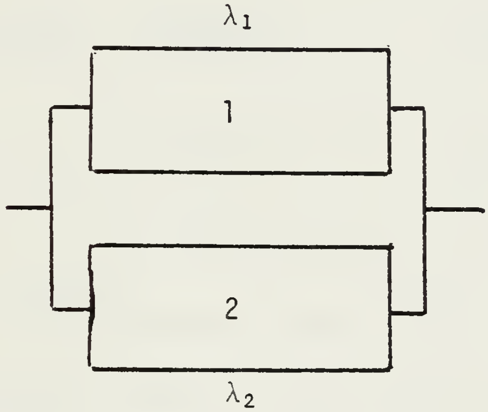
The ZERO distribution contributes zero additional life to the system, so we can omit it. Nevertheless, we can not omit its probability in the calculation of path probability, since the event numbered 8 has a probability of occurring.

B. THE SURVIVAL FUNCTION FOR THE SYSTEM

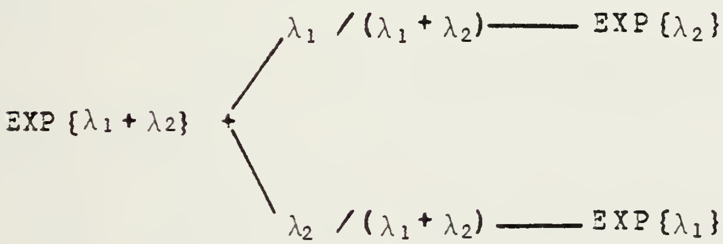
In this section we will deal with an approach to obtaining the system's survival function.

1. EXAMPLES

SYSTEM: Two components in parallel.



LIFE :



DESCRIPTION: Component 1 has EXP$\{\lambda_1\}$ life and component 2 has EXP$\{\lambda_2\}$ life.

There are two failure events that can be allowed. The first is that component 1 fails at some time t and component 2 carries the system for the rest of the time. The other one

is that component 2 fails at some time during the mission and component 1 carries the system for the rest of the time. There are more events such as no failures during the mission duration, which are taken care of by the representation.

Now we have two paths

No.of Path	Weight	Life
1	$p_1 = \lambda_1/(\lambda_1+\lambda_2)$	$EXP\{\lambda_1+\lambda_2\}+EXP\{\lambda_2\}$
2	$p_2 = \lambda_2/(\lambda_1+\lambda_2)$	$EXP\{\lambda_1+\lambda_2\}+EXP\{\lambda_1\}$

Let T be the system's time to failure and let T1,T2,T3 be random variables exponentially distributed with the failure rates $\lambda_1+\lambda_2, \lambda_2, \lambda_1$ respectively.

Then

$$T = \begin{cases} T1 + T2 & \text{with probability } p_1 \\ T1 + T3 & \text{with probability } p_2 \end{cases}$$

The survival function can be written as

$$\bar{F}(t) = p_1 \bar{F}_1(t) + p_2 \bar{F}_2(t) \quad , t \geq 0$$

where $p_1 = \lambda_1/(\lambda_1+\lambda_2)$, $p_2 = \lambda_2/(\lambda_1+\lambda_2)$ and $\bar{F}_1(t)$, $\bar{F}_2(t)$ denote the survival functions for the shorthand notations $EXP\{\lambda_1+\lambda_2\}+EXP\{\lambda_2\}$, $EXP\{\lambda_1+\lambda_2\}+EXP\{\lambda_1\}$ respectively.

The survival function for the convolution of two exponentially distributed random variables with dissimilar failure rates is

$$\bar{F}(t)=\lambda_2/(\lambda_2-\lambda_1)\ e^{-\lambda_1 t}+\lambda_1/(\lambda_1-\lambda_2)\ e^{-\lambda_2 t}\ t\geq 0.$$

If we do the substitutions for $\bar{F}_1(t)$ and $\bar{F}_2(t)$ as $\lambda_1=\lambda_1+\lambda_2$, $\lambda_2=\lambda_2$ and $\lambda_1=\lambda_1+\lambda_2$, $\lambda_2=\lambda_1$ respectively, $\bar{F}_1(t)$ becomes

$$\bar{F}_1(t)=\lambda_2/(-\lambda_1)\ e^{-(\lambda_1+\lambda_2)t}+(\lambda_1+\lambda_2)/\lambda_1\ e^{-\lambda_2 t}\ ,\ t\geq 0,$$

and $\bar{F}_2(t)$ becomes

$$\bar{F}_2(t)=\lambda_1/(-\lambda_2)\ e^{-(\lambda_1+\lambda_2)t}+(\lambda_1+\lambda_2)/\lambda_2\, e^{-\lambda_2 t}\ ,\ t\geq 0.$$

Then

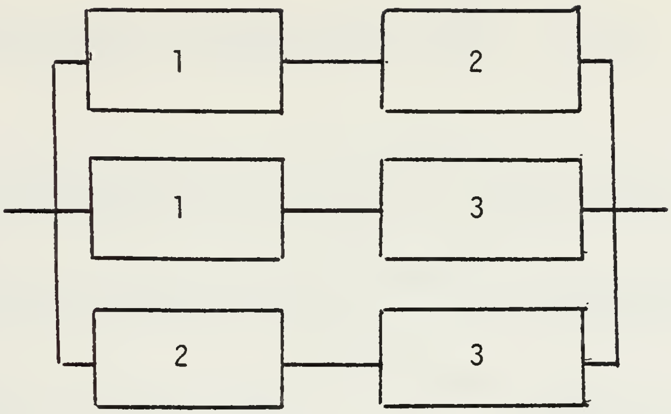
$$\bar{F}(t)=\lambda_1/(\lambda_1+\lambda_2)\ \bar{F}_1(t)\ +\lambda_2/(\lambda_1+\lambda_2)\ .F\ (t)$$

$$\bar{F}(t)=e^{\lambda_1}\ +e^{\lambda_1}\ -e^{(\lambda_1+\lambda_2)t}\ ,\ t\geq 0.$$

The result gives the survival function that is well known for this system.

Another example is the 2 out of 3 system.

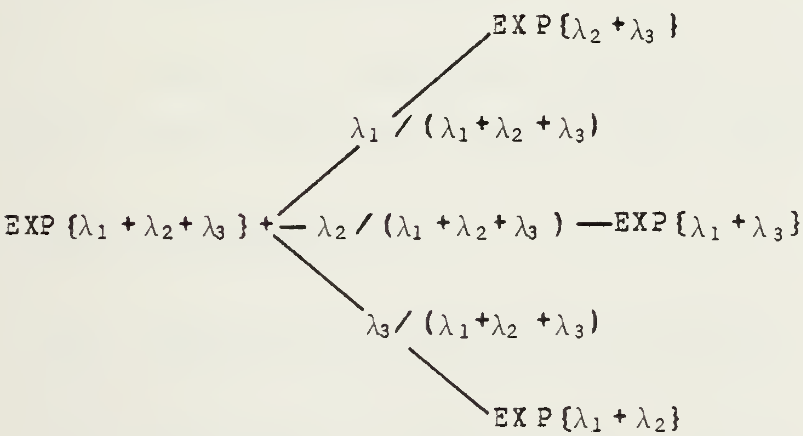
RELIABILITY NETWORK:



The known survival function is

$$\overline{F}(t)=e^{-(\lambda_1+\lambda_2)t}+e^{-(\lambda_1+\lambda_3)t}+e^{-(\lambda_2+\lambda_3)t}-2e^{-(\lambda_1+\lambda_2+\lambda_3)t}\quad,\ t\geq 0.$$

LIFE:



$$T=\begin{cases} EXP\{\lambda_1+\lambda_2+\lambda_2\}+EXP\{\lambda_2+\lambda_3\} & \text{with probability } \lambda_1/(\lambda_1+\lambda_2+\lambda_3) \\ EXP\{\lambda_1+\lambda_2+\lambda_3\}+EXP\{\lambda_1+\lambda_3\} & \text{with probability } \lambda_2/(\lambda_1+\lambda_2+\lambda_3) \\ EXP\{\lambda_1+\lambda_2+\lambda_3\}+EXP\{\lambda_1+\lambda_2\} & \text{with probability } \lambda_2/(\lambda_1+\lambda_2+\lambda_3) \end{cases}$$

The survival function is

$$\overline{F}(t)=p_1\overline{F}_1(t)+p_2\overline{F}_2(t)+p_3\overline{F}_3(t) \quad , \ t\geq 0.$$

where

$$\overline{F}_1(t)=(\lambda_2+\lambda_3)/(-\lambda_1)\ e^{-(\lambda_1+\lambda_2+\lambda_3)t}+(\lambda_1+\lambda_2+\lambda_3)/\lambda_1\ e^{-(\lambda_2+\lambda_3)t}$$

$$\overline{F}_2(t)=(\lambda_1+\lambda_3)/(-\lambda_2)\ e^{-(\lambda_1+\lambda_2+\lambda_3)t}+(\lambda_1+\lambda_2+\lambda_3)/\lambda_2\ e^{-(\lambda_1+\lambda_3)t}$$

$$\overline{F}_3(t)=(\lambda_1+\lambda_2)/(-\lambda_3)\ e^{-(\lambda_1+\lambda_2+\lambda_3)t}+(\lambda_1+\lambda_2+\lambda_3)/\lambda_3\ e^{-(\lambda_1+\lambda_2)t}$$

If we do the necessary cancellations, we can get,

$$\overline{F}(t)=e^{-(\lambda_2+\lambda_3)t}+e^{-(\lambda_1+\lambda_3)t}+e^{-(\lambda_1+\lambda_2)t}-2e^{-(\lambda_1+\lambda_2+\lambda_3)t}, \ t\geq 0.$$

as desired.

2. General Procedure

Having the algorithm presented in Section II, we can treat more complicated systems under similar assumptions.

The procedure is

i. Set up the reliability network for the system.

ii. According to this network, set up the system life.

iii. Using the proper reliability shorthand formulas for the related convolutions of exponential lives, set up the survival function.

IV. SUMMARY

The reliability shorthand is an easy way to describe a system's life, but it is difficult to implement computatio-naly since there is considerable complexity in handling con-volutions.

The algorithm presented in this paper gives some relief from this difficulty. However, the accuracy in obtained from this algorithm is very much related to the differences in the failure rates.

Another aspect in the algorithm is that distributions are convolved one at a time and this requires very accurate run-ning conditions in the case of a complicated system.

It is believed that it is possible to derive another algorithm which is more powerful than the one introduced here. Instead of adding one distribution at a time, one can try to add several at a time.

APPENDIX A

A-1

This section contains the survival functions for several shorthand notations which were derived by the use of the approach described in Section II.

A.1.1 Shorthand Notation : EXP{ λ }.

Survival Function: $\overline{F}(t) = e^{-\lambda t}$, $t \geq 0$.

A.1.2.1 Shorthand Notation: EXP{λ_1}+ EXP{λ_2}

Survival Function;

$$\overline{F}(t) = \lambda_2/(\lambda_2-\lambda_1)\, e^{-\lambda_1} + \lambda_1/(\lambda_1-\lambda_2)\, e^{-\lambda_2 t} \quad ,t\geq 0.$$

A.1.2.2 Shorthand Notation : EXP{ λ }+EXP{ λ }.

Survival Function:

$$\overline{F}(t) = (1+\lambda t)\, e^{-\lambda t} \quad , t \geq 0.$$

A.1.3.1 Shorthand Notation : EXP{λ_1}+EXP{λ_2}+EXP{λ_3 }

Survival Function:

$$\overline{F}(+) = a_{11}\, e^{-\lambda_1 t} + a_{21}\, e^{-\lambda_2 t} + a_{31}\, e^{-\lambda_3 t} \quad , t\geq 0.$$

where $a_{11} = \lambda_2\lambda_3/(\lambda_2-\lambda_1)(\lambda_3-\lambda_1)$, $a_{21} = \lambda_1\lambda_3/(\lambda_1-\lambda_2)(\lambda_3-\lambda_2)$,

$a_{31}=\lambda_1\lambda_2/(\lambda_1-\lambda_3)(\lambda_2-\lambda_3)$.

A.1.3.2 EXP{λ}+EXP{λ}+EXP{λ}

Survival Function:

$$\bar{F}(t)=(1+\lambda t+1/2\lambda^2t^2)e^{-\lambda t}\quad,\ t\geq 0.$$

A.1.3.3 EXP{λ_1}+EXP{λ_2}+EXP{λ_2}

Survival Function:

$$\bar{F}(t)=a_{11}e^{-\lambda_1 t}+(a_{21}+a_{22}t)e^{-\lambda_2 t}\quad,\ t\geq 0$$

where $a_{11}=\lambda_2^2/(\lambda_2-\lambda_1)^2$, $a_{21}=(\lambda_1^2-2\lambda_1\lambda_2)/(\lambda_1-\lambda_2)^2$

$a_{22}=\lambda_1\lambda_2/(\lambda_1-\lambda_2)$.

A.1.4.1 EXP{λ_1}+EXP{λ_2}+EXP{λ_3}+EXP{λ_4}

Survival Function:

$$\bar{F}(t)=a_{11}e^{\lambda_1}+a_{21}e^{-\lambda_2 t}+a_{31}e^{-\lambda_3 t}+a_{41}e^{-\lambda_4 t}\quad,\ t\geq 0.$$

where $a_{i1}=\prod_{j\neq i}\lambda j/\prod_{j\neq i}(\lambda_j-\lambda_i)\quad\forall\ i=1,2,3,4$

A.1.4.2 EXP{λ}+EXP{λ}+EXP{λ}+EXP{λ}

Survival Function:

$$\bar{F}(t)=(1+\lambda t+1/2\,\lambda^2t^2+1/6\,\lambda^3t^3)e^{-\lambda t}\quad,\ t\geq 0.$$

A.1.4.3 EXP{λ_1}+EXP{λ_2}+EXP{λ_2}+EXP{λ_2}

Survival Function:

$$\bar{F}(t)=a_{11}e^{-\lambda_1 t}+(a_{21}+a_{22}t+a_{23}t^2)e^{-\lambda_2 t},\ t\geq 0.$$

where $a_{11}=\lambda_2{}^3/(\lambda_2-\lambda_1)^3$, $a_{21}=1-a_{11}$,

$a_{22}=\lambda_1\lambda_2(\lambda_1-2\lambda_2)/(\lambda_1-\lambda_2)^2$, $a_{23}=\lambda_1\lambda_2{}^2/2(\lambda_1-\lambda_2)$.

A.1.4.4 EXP{λ_1}+EXP{λ_1}+EXP{λ_2}+EXP{λ_2}

Survival Function:

$$\bar{F}(t)=(a_{11}+a_{12}t)e^{-\lambda_2 t}+(a_{21}+a_{22}t)e^{-\lambda_2 t},\ t\geq 0.$$

where $a_{11}=(\lambda_2{}^3-3\lambda_2^2\lambda_1)/(\lambda_2-\lambda_1)^3$, $a_{12}=\lambda_1\lambda_2^2/(\lambda_2-\lambda_1)^2$,

$a_{21}=(\lambda_1{}^3-3\lambda_1^2\lambda_2)/(\lambda_1-\lambda_2)^3$, $a_{22}=\lambda_2\lambda_1{}^2/(\lambda_1-\lambda_2)^2$.

A.1.4.5 EXP{λ_1}+EXP{λ_2}+EXP{λ_3}+EXP{λ_3}

Survival Function:

$$\bar{F}(t)=a_{11}e^{-\lambda_1 t}+a_{21}e^{-\lambda_2 t}+(a_{31}+a_{32}t)e^{-\lambda_3 t},\ t\geq 0.$$

where $a_{11}=\lambda_2\lambda_3{}^2/(\lambda_2-\lambda_1)(\lambda_3-\lambda_1)^2$, $a_{21}=\lambda_1\lambda_2^2/(\lambda_1-\lambda_2)(\lambda_3-\lambda_2)^2$,

$$a_{31}=\lambda_1\lambda_2/(\lambda_1-\lambda_3)(\lambda_2-\lambda_3)+\lambda_1\lambda_2\lambda_3[1/(\lambda_1-\lambda_2)(\lambda_1-\lambda_3)^2$$
$$-1/(\lambda_1-\lambda_2)(\lambda_2-\lambda_3)^2]$$

$a_{32}=\lambda_1\lambda_2\lambda_3/(\lambda_1-\lambda_3)(\lambda_2-\lambda_3)$.

This section introduces a Fortran program using the algorithm described in Section II.

A PROGRAM FOR THE ALGORITHM ONE AT A TIME:

```
C
C   THIS IS A PROGRAM TO COMPUTE THE RELIABILITY OF A SYSTEM
C       WHICH HAS THE RELIABILITY SHORTHAND NOTATION
C                   EXP(L1)+..+EXP(LN)
C    WHERE THERE IS NO RESTRICTION FOR THE FAILURE RATES.
C
C         VARIABLES:
C   A(I,J)    : I.TH TYPE FAILURE RATE, J.TH COEFFICIENT
C                        ON THE POLINOM.
C   NI(I)     : AMOUNT OF LIVES HAVING THE I.TH TYPE
C                          FAILURE RATE.
C   NINIT(I)  : AUXILAURY ARRAY FOR THE NI(I). INPUT
C                         FOR THE PROGRAM.
C    L(I)     : THE ARRAY FOR THE FAILURE RATES
C
      REAL L(20),A(20,10)
      INTEGER NINIT(20),NI(20)
C
C                                              GET INPUT
C
      READ(5,499) T
      WRITE(6,498) T .
      READ(5,500) K
      WRITE(6,1500) K
      READ(5,501)  (NINIT(I),L(I),I=1,K)
      WRITE(6,1501)  (NINIT(I),L(I),I=1,K)
C
C                              COMPUTE COEFFICIENTS ONE ALL
C                                         DISSIMILAR
      KK=1
    2 A(KK,1)=1.
      NI(KK)=1
            JJ=1
    4       IF(JJ.EQ.KK) GO TO 6
            A(KK,1)=A(KK,1)*L(JJ)/(L(JJ)-L(KK))
    6     IF(JJ.EQ.K) GO TO 9
            JJ=JJ+1
            GO TO 4
    9 IF(KK.EQ.K) GO TO 8
      KK=KK+1
      GO TO 2
    8 CONTINUE
C
C                                  BEGIN TO ADD ONE AT A TIME
C
      IE=1
   32 IF(NINIT(IE).EQ.NI(IE)) GO TO 99
      NI(IE)=NI(IE)+1
C
C                                             UPDATE  IE
```

```
C
       J=0
 12    NNNN=NI(IE)-J
       A(IE,NNNN)=L(IE)*A(IE,NNNN-1)/FLOAT(NNNN-1)
       IF(NNNN.EQ.2) GO TO 20
       J=J+1
       GO TO 12
 20 SUM=0.
          DO 101 I=1,K
          IF(I.EQ.IE) GO TO 101
          SUM=SUM+A(I,1)*L(I)/(L(I)-L(IE))
          IF(NI(I).LT.2) GO TO 101
          FACT=1.
          NKK=NI(I)
                DO 102 II=2,NKK
                FACT=FACT*(II-1)
        SUM=SUM+A(I,II)*FACT*L(IE)/(L(I)-L(IE))**II
102             CONTINUE
101       CONTINUE
          A(IE,1)=SUM+A(IE,1)
C
C                                          UPDATE  I.NE.IE
C
    I=1
 21 IF(I.EQ.IE) GO TO 26
    A(I,NI(I))=A(I,NI(I))*L(IE)/(L(IE)-L(I))
       IF(NI(I).EQ.1) GO TO 26
          J=1
 24       NNI=NI(I)-J
          A(I,NNI)=(L(IE)*A(I,NNI)-FLOAT(NNI)*A(I,NNI+1))
    *                /(L(IE)-L(I))
          IF(J.EQ.(NI(I)-1)) GO TO 26
          J=J+1
          GO TO 24
 26 IF(I.EQ.K) GO TO 32
    I=I+1
    GO TO 21
 99 CONTINUE
    IF(IE.EQ.K) GO TO 35
    IE=IE+1
    GO TO 32
C
C                             CALCULATE THE PROBABILITY
C
 35 CONTINUE
    PRO=0.
    DO 40 I=1,K
    SUM=0.
    NKK=NI(I)
       DO 41 J=1,NKK
       SUM=SUM+A(I,J)*T**(J-1)
 41    CONTINUE
    PRO=PRO+SUM*EXP(-L(I)*T)
 40 CONTINUE
C
C                                     PRINT OUT THE RESULTS
C
    WRITE(6,600)
    DO 17 I=1,K
    NNK=NI(I)
    WRITE(6,1600) L(I),NI(I)
 17 WRITE(6,603) (A(I,J),J=1,NNK)
    WRITE(6,601) T,PRO
    STOP
498 FORMAT('1',10X,'THE INPUT IS:',//15X,'TIME=',F10.4)
499 FORMAT(F10.4)
500 FORMAT(3X,I3)
501 FORMAT(3X,I3,F10.5)
```

```
 600 FORMAT('1',10X,'FAILURE RATE      AMOUNT  COEFFICIENTS
    *',/10X,40('-'))
 601 FORMAT('0',10X,'PROBABILITY ( TIME > ',F10.5,')= ',
    *F10.7)
 603 FORMAT('0',35X,3(5(E12.5,2X),/))
1500 FORMAT('0',14X,'K   =',3X,I2,7X,'(THE NUMBER OF
    *DISSIMILAR FAILURE RATES)')
1501 FORMAT('0',15X,'# OF R.V.    FAILURE RATE',/15X,
    125('-'),//20(18X,I3,5X,F10.5,//))
1600 FORMAT('0',10X,F10.5, 5X,I3)
     END
```

EXAMPLE :

<u>INPUT FOR THE PROGRAM</u> :

```
       8.
K=      6
        5   6.5
        5   1.5
        5   2.5
        5   3.5
        5   5.0
        5   9.0
```

<u>OUTPUT FROM THE PROGRAM</u> :

```
   THE INPUT IS:
    TIME=     8.0000
    K    =    6         (THE NUMBER OF DISSIMILAR LAMDA" S)
    NO. OF R.V.      FAILURE RATE
    ----------       ------------
          5          6.5
          5          1.5
          5          2.5
          5          3.5
          5          5.0
          5          9.0
  FAILURE RATE       AMOUNT   COEFFICIENTS
 --------------     -------   ------------
      6.5              5
              0.15713E+08   0.84332E+07    0.18530E+07
              0.20038E+06   0.92701E+04
      1.5              5
              0.27364E+09  -0.89949E+08    0.11817E+08
             -0.73813E+06   0.18586E+05
      2.5              5
             -0.18295E+11   0.82312E+10   -0.17601E+10
              0.17341E+09  -0.11979E+08
      3.5              5
              0.18841E+11   0.89164E+10    0.25648E+10
              0.25077E+09   0.33406E+08
      5.0              5
             -0.83480E+09  -0.52324E+09   -0.16061E+09
             -0.21713E+08  -0.22945E+07
      9.0              5
             -0.39700E+05  -0.15168E+06   -0.32323E+06
             -0.33727E+06  -0.27860E-01

    PROBABILITY ( TIME > 8.0 )=   0.6935042
```

APPENDIX B

This section contains a program described in Section III to compute the reliability of a system.

```
C
C
C       THIS PROGRAM CALCULATES  THE RELIABILITY FUNCTION:
C
C
C       F(T) =
C    P1   *   F1(T) (=EXP(L1,1) +EXP(L1,2) +........+EXP(L1,M1))
C  + P2   *   F2(T) (=EXP(L2,1) +EXP(L2,2) +........+EXP(L2,M2))
C  + P3   *   F3(T) (=EXP(L3,1) +EXP(L3,2) +........+EXP(L3,M3))
C  +  . . . . . . . . . . . . . . . . . . . . . . . .
C     . . . . . . . . . . . . . . . . . . . . . . . . .
C  +  . . . . . . . . . . . . . . . . . . . . . . . .
C  + PI   *   FI(T) (=EXP(LI,1) +EXP(LI,2) +........+EXP(LI,MI))
C  +  . . . . . . . . . . . . . . . . . . . . . . . .
C  +  . . . . . . . . . . . . . . . . . . . . . . . .
C     . . . . . . . . . . . . . . . . . . . . . . . . .
C  +  . . . . . . . . . . . . . . . . . . . . . . . .
C  + PN   *   FN(T) (=EXP(LN,1) +EXP(LN,2) +........+EXP(LN,MN))
C
C       L( )      : ARRAY FOR ALL LAMDAS.
C       P( )      : ARRAY FOR ALL PROBABILITIES.
C       MI( )     : NUMBER OF EXP IN EACH ROW.
C       IK        : TOTAL NUMBER OF ROWS.
C       PROBA     : PROBABILITY OF SYSTEM SURVIVAL AT TIME T.
C       TIME      : TIME
C       A(I,K)    : THE COEFFICIENT FOR THE CURRENT PATH,
C                      I.TH TYPE FAILURE
C             RATE AND K.TH COEFFICIENT ON THE THIS POLINOMIAL
C
      REAL P(50)
      CALL READ1(IK,P,T)
      SUM=0.
         DO 1 I=1,IK
    1    SUM=SUM+P(I)
      IF(ABS(SUM-1.).GT.1.0E-5) GO TO 199
      PROBA=0.0
         DO 2 I=1,IK
         WRITE(6,1009) I,I,P(I)
         CALL ONECON(PRO,T)
    2    PROBA=PROBA+PRO*P(I)
      WRITE(6,601) T,PRO
      STOP
  199 WRITE(6,198) SUM
      STOP
  601 FORMAT('1',10X,'PROBABILITY ( TIME > ',F10.5,')= ',
     *F10.7)
 1009 FORMAT('0',10X,I2,'.',3X,'P(',I2,')=',F10.7)
  198 FORMAT('1',10X,'THE SUM OF THE PROBABILITIES IS NOT
     * EQUAL TO 1.0',/10X,'SUM = ',F10.7)
      END
C
C                                                GET INPUT
C
      SUBROUTINE READ1(IK,P,T)
      REAL P(50)
      READ(5,499) T
```

```
      WRITE(6,1499) T
      READ(5,500) IK
      WRITE(6,1500) IK
      READ(5,3)  (P(I),I=1,IK)
      RETURN
    3 FORMAT(7F10.7)
  499 FORMAT(5X,F10.4)
  500 FORMAT(5X,I5)
 1499 FORMAT('1',10X,'TIME=',F10.4)
 1500 FORMAT('0',10X,'IK  ='3X,I3)
      END
C
C
C
      SUBROUTINE ONECON(PRO,T)
      REAL A(20,10),L(20)
      INTEGER NI(20),NINIT(20)
      CALL READ(K,L,NINIT)
      CALL ONEDIS(K,A,NI,L)
      CALL ONEATA(K,A,NI,L,NINIT)
      CALL CALPRO(K,A,NI,L,PRO,T)
      RETURN
      END
C
C
C
      SUBROUTINE READ(K,L,NINIT)
      REAL L(20)
      INTEGER NINIT(20)
      READ(5,1) K
      WRITE(6,1501) K
      READ(5,52)  (L(I),NINIT(I),I=1,K)
      RETURN
    1 FORMAT(5X,I5)
   52 FORMAT(5X,F10.5,I5)
 1501 FORMAT(10X,'K   =',I2,5X,'(# OF DISSIMILAR FAILURE
     * RATES.)')
      END
C
      SUBROUTINE ONEDIS(K,A,NI,L)
      REAL A(20,10),L(20)
      INTEGER NI(20)
      KK=1
    2 A(KK,1)=1.
      NI(KK)=1
      JJ=1
    4 IF(JJ.EQ.KK) GO TO 6
      A(KK,1)=A(KK,1)*L(JJ)/(L(JJ)-L(KK))
    6 IF(JJ.EQ.K) GO TO 9
      JJ=JJ+1
      GO TO 4
    9 IF(KK.EQ.K ) GO TO 8
      KK=KK+1
      GO TO 2
    8 RETURN
      END
C
C                                   BEGIN TO ADD ONE AT A TIME
C
      SUBROUTINE ONEATA(K,A,NI,L,NINIT)
      REAL L(20),A(20,10)
      INTEGER NINIT(20),NI(20)
      IE=1
   32 IF(NINIT(IE).EQ.NI(IE)) GO TO 99
      NI(IE)=NI(IE)+1
C
C                                        UPDATE  IE
C
```

```
      J=0
   12 NN=NI(IE)-J
      A(IE,NN)=L(IE)*A(IE,NN-1)/FLOAT(NN-1)
      IF(NN.EQ.2) GO TO 20
      J=J+1
      GO TO 12
   20 CONTINUE
         DO 101 I=1,K
         IF(I.EQ.IE) GO TO 101
         A(IE,1)=A(IE,1) +A(I,1)*L(I)/(L(I)-L(IE))
         IF(NI(I).LT.2) GO TO 101
         FACT=1.
         NKK=NI(I)
               DO 102 II=2,NKK
               FACT=FACT*(II-1)
     A(IE,1)=A(IE,1)+A(I,II)*FACT*L(IE)/(L(I)-L(IE))**II
  102          CONTINUE
  101    CONTINUE
C
C                                               UPDATE  I.NE.IE
C
      I=1
   21 IF(I.EQ.IE) GO TO 26
      A(I,NI(I))=A(I,NI(I))*L(IE)/(L(IE)-L(I))
         IF(NI(I).EQ.1) GO TO 26
         J=1
   24    NNI=NI(I)-J
     A(I,NNI)=(L(IE)*A(I,NNI)-FLOAT(NNI)*A(I,NNI+1))
     *              /(L(IE)-L(I))
         IF(J.EQ.(NI(I)-1)) GO TO 26
         J=J+1
         GO TO 24
   26 IF(I.EQ.K) GO TO 32
      I=I+1
      GO TO 21
   99 CONTINUE
      IF(IE.EQ.K) GO TO 35
      IE=IE+1
      GO TO 32
   35 RETURN
      END
C
C
C
      SUBROUTINE CALPRO(K,A,NI,L,PRO,T)
      REAL L(20),A(20,10)
      INTEGER NI(20)
      PRO=0.
      DO 40 I=1,K
      SUM=0.
      NKK=NI(I)
         DO 41 J=1,NKK
         SUM=SUM+A(I,J)*T**(J-1)
   41    CONTINUE
      TTT=-L(I)*T
      PRO=PRO+SUM*EXP(TTT)
   40 CONTINUE
      WRITE(6,1010)
            DO 1000 I=1,K
            NNN=NI(I)
            WRITE(6,1001) L(I),NI(I),(A(I,J),J=1,NNN)
 1000       CONTINUE
      RETURN
 1001 FORMAT(10X,F10.5,5X,I3,5X,,2(5(E12.5,5X),/30X))
 1010 FORMAT(15X,'LAMDA ',6X,'NI',5X,'COEFFICIENTS',/14X,
     135('-'))
      END
```

INPUT FOR THE PROGRAM :

```
TIME      15.
#ROWS     5
  .2          .2          .2          .2          .2
1.ST      3
          .1      3
          .8      2
          .4      1
 2.ND     3
          .2      2
          .5      4
          .6      1
 3.RD     3
          .1      3
          .5      4
          .3      2
 4.TH     2
          .1      3
          .2      3
 5.TH     3
          .1      2
          .2      4
          .4      3
```

OUTPUT FROM THE PROGRAM :

```
 TIME=    15.0000
 IK   =      5
 1.     P(1) = 0.2
K    = 3       (# OF DISSIMILAR FAILURE RATES.)
LAMDA   NI   COEFFICIENTS
-----   ---  ------------
0.10    3    0.11294E+01 0.66342E-01 0.87075E-02
0.80    2    0.18742E-01 0.23324E-02
0.40    1   -0.14815E+00
 2.     P(2) = 0.2
K    = 3       (# OF DISSIMILAR FAILURE RATES.)
LAMDA   NI   COEFFICIENTS
-----   ---  ------------
0.20    2   -0.25077E+02 0.23148E+01
0.50    4   -0.13017E+03 0.21333E+02 -0.22222E-01 0.55556E-01
0.60    1    0.15625E+03
 3.     P(3) = 0.2
  K     = 3       (# OF DISSIMILAR FAILURE RATES.)
LAMDA   NI  COEFFICIENTS
-----   --- ------------
0.10    3    0.75531E+01 -0.54932E+00  0.27466E-01
0.50    4   -0.89945E+01 -0.99536E+00 -0.42847E-01 -0.73242E-03
0.30    2    0.24414E+01 -0.14648E+01
 4.     P( 4) = 0.2
  K     = 2       (# OF DISSIMILAR FAILURE RATES.)
LAMDA   NI  COEFFICIENTS
-----   --- ------------
0.10    3  0.32000E+02 -0.16000E+01  0.40000E-01
0.20    3 -0.31000E+02 -0.14000E+01 -0.20000E-01
 5.     P( 5) = 0.2
  K     = 3       (# OF DISSIMILAR FAILURE RATES.)
LAMDA   NI  COEFFICIENTS
-----   --- ------------
0.10    2 -0.15170E+03  0.37926E+01
0.20    4  0.14400E+03  0.12800E+02 0.32000E+00 0.10667E-01
0.40    3  0.87037E+01  0.51851E+00 0.88889E-02
   PROBABILITY ( TIME > 15.0 )=   0.9111188
```

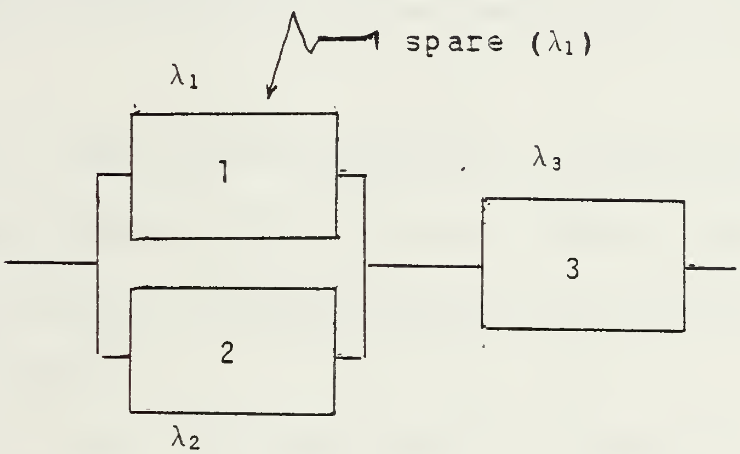
APPENDIX C

INTRODUCTION

This section consists of a computer program to compute system reliabilities as described in Section III.

Again for simplicity, we will go thruogh an example.

The structure of the system is



Components 1, 2 and 3 have lives exponentially distributed with failure rates λ_1 ,λ_2 , λ_3 respectively. Also we have a spare for component 1 which is identical to component 1.

Using the shorthand approach, the system life would be determined as follows

$\lambda_1 / (\lambda_1+\lambda_2+\lambda_3)$ — EXP $\{\lambda_2+\lambda_3\}$

EXP $\{\lambda_1+\lambda_2+\lambda_3\}$ + — $\lambda_2 / (\lambda_1+\lambda_2+\lambda_3)$ — EXP $\{\lambda_1+\lambda_3\}$

$\lambda_1 / (\lambda_1+\lambda_2+\lambda_3)$ $\qquad$ $\lambda_3 / (\lambda_1+\lambda_2+\lambda_3)$ — ZERO

EXP $\{\lambda_1+\lambda_2+\lambda_3\}$ + $\lambda_3 / (\lambda_1+\lambda_2+\lambda_3)$ — ZERO

$\lambda_2 / (\lambda_1+\lambda_2+\lambda_3)$ $\qquad$ $\lambda_1 / (\lambda_1+\lambda_3)$ — EXP $\{\lambda_1+\lambda_3\}$

EXP $\{\lambda_1+\lambda_3\}$ +

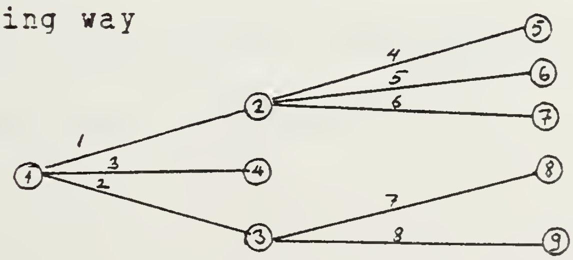
$\lambda_3 / (\lambda_1+\lambda_3)$ — ZERO

Some definitions are necessary before describing the program

<u>NODE</u>: Each node represents an exponentially distrubuted random variable with a certain failure rate. The number for any node can be chosen arbitrarily but can not be used more than once.

<u>ARC</u> : Each arc originates at a node and leads to another node. Only one arc can terminate at a given node. Arc numbers can be chosen arbitrarily. If n is the total number of nodes in a system life then there will be n-1 arcs in this system.

With these definitions, we can represent a system life in the following way



BACK POINTER LIST (IPB): Each node has a back pointer. A back pointer is an arc number which shows which arc connects the node to the tree. If the pointer is zero, then the related node is the root of the tree.

NODE NO	BACK POINTER IPB(I)
1	0
2	1
3	2
4	3
5	4
6	5
7	6
8	7
9	8

(Here node 1 is the root of the tree.)

NODE CODE LIST: As we mentioned, each node represents an exponential lifetime, for simplicity we can use some integer code numbers for each failure rate.

Code No.	Related Failure Rate-L(I)
0	(distribution ZERO)
1	$\lambda_1 + \lambda_2 + \lambda_3$
2	$\lambda_1 + \lambda_3$
3	$\lambda_1 + \lambda_2$

Here only the code number 0 is not arbitrary and 0 can be used for the ZERO distribution. The others can be picked out arbitrarily.

ARC CODE LIST: This list is similar to the node code list. Arc code numbers represent probabilities.

Code No.	Probability-PA(I)
1	$\lambda_1 / (\lambda_1 + \lambda_2 + \lambda_3)$
2	$\lambda_2 / (\lambda_1 + \lambda_2 + \lambda_3)$
3	$\lambda_3 / (\lambda_1 + \lambda_2 + \lambda_3)$
4	$\lambda_1 / (\lambda_1 + \lambda_3)$
5	$\lambda_3 / (\lambda_1 + \lambda_3)$

ARC ORIGIN LIST : Each arc has an origin node and a terminal node. In the program we need to use only the origin list.

Arc No.	Origin Node IO(I)
1	1
2	1
3	1
4	2
5	2
6	2
7	3
8	3

LAST POINT NODE LIST (LP): This list indicates the nodes on the end of each path. There is no necessary order in the list.

LAST POINT DEPTH LIST (IPD): This list indicates the number of arcs from last node to the root of the tree.

I	L. P. Node List LP(I)	L. P. Depth List IPD(I)
1	5	2
2	6	2
3	7	2
4	8	2
5	9	2
6	4	1

```
C
C     THIS PROGRAM COMPUTES THE RELIABILITY OF A SYSTEM
C     WHICH HAS A SURVIVAL FUNCTION LIKE BELOW:
C
C     F(T)=
C   P1  *  F1(T) (=EXP(L1,1) +EXP(L1,2) +.......+EXP(L1,M1))
C +P2   *  F2(T) (=EXP(L2,1) + EXP(L2,2) +.......+EXP(L2,M2))
C +P3   *  F1(T) (=EXP(L3,1) + EXP(L3,2) +.......+EXP(L3,M3))
C +   . . . . . . . . . . . . . . . . . . . .
C .   . . . . . . . . . . . . . . . . . . . .
C +   . . . . . . . . . . . . . . . . . . . .
C +  PI   *  FI(T) (=EXP(LI,1) +EXP(LI,2) +.......+EXP(LI,MI))
C +   . . . . . . . . . . . . . . . . . . . .
C +   . . . . . . . . . . . . . . . . . . . .
C   . . . . . . . . . . . . . . . . . . . . .
C +   . . . . . . . . . . . . . . . . . . . .
C +  PN   *  FN(T) (=EXP(LN,1) +EXP(LN,2) +.......+EXP(LN,MN))
C
C
C            VARIABLES IN PROGRAM:
C   IPB(I)     : BACK POINTER OF THE I.TH NODE
C   IPA(K)     : PROBABILITY CODE OF THE K.TH ARC
C   IO(K)      : ORIGIN NODE OF K.TH ARC
C   ICL(I)     : FAILURE RATE CODE OF THE I.TH NODE
C   PA()       : DIFFERENT PROBABILITY LIST
C   LP()       : LAST POINT NODE LIST
C   IPD()      : LAST POINT NODE DEPTH LIST
C   L()        : FAILURE RATE LIST
C   N          : NUMBER OF NODES IN TREE
C   M          : NUMBER OF ARCS IN TREE
C   NLAST      : NUMBER OF LAST POINT NODES
C   NDIF       : NUMBER OF DIFFERENT FAILURE RATES
C   MDIF       : NUMBER OF DIFFERENT PROBABILITIES
C   NPATH()    : NODE LIST IN A PARTICULAR PATH
C   MPATH()    : ARC LIST IN A PARTICULAR PATH
C   IM         : THE NUMBER OF THE CURRENT LAST POINT NODE
C   IP         : THE DEPTH OF THE CURRENT LAST POINT NODE
C   PROBA      : THE CURRENT VALUE OF THE SURVIVAL FUNCTION
C   PRO        : PROB. OF THE CURRENT PATH SURVIVAL AT T
C   P          : THE CURRENT PATH PROBABILITY
C   LCODE()    : FAILURE RATE CODES IN CURRENT PATH
C   LL()       : FAILURE RATE LIST IN CURRENT PATH
C   IK         : # OF DIS. FAILURE RATES IN CURRENT PATH
C   NI()       : NUMBER OF LIFES RELATED TO THE FAILURE RATE
C   NINIT()    : AUXILAURY ARRAY FOR NI()
C   A(,)       : COEFFICIENTS FOR CURRENT PATH
C   T          : TIME
C
C
C
C                          MAIN PROGRAM
C
C
      REAL L(20),PA(20),LL(11),PRO,P,PROBA,T
      INTEGER IPB(100),ICL(100),IO(100),LP(50),IPD(50),
     *NPATH(11),MPATH(10),NI(11),I,IK,IPA(100)
      PROBA=0.0
      CALL READ(LP,L,PA,IPB,ICL,IO,IPD,NLAST,T,IPA)
      WRITE(6,1013)
         DO 1 I=1,NLAST
         IM=LP(I)
         IP=IPD(I)
      NPATH(1)=IM
      DO 9 J=1,IP
      NPATH(J+1)=IO(IPB(IM))
      MPATH(J)=IPB(IM)
      IM=NPATH(J+1)
    9 CONTINUE
```

```
          CALL PASS(PA,L,ICL,NPATH,MPATH,IP,LL,NI,IK,P,IPA)
          WRITE(6,1990) I,I,P
          CALL ONECON(PRO,T,LL,NI,IK)
          PROBA=PROBA+P*PRO
    1     CONTINUE
      WRITE(6,601) T,PROBA
      STOP
  601 FORMAT('0',10X,'PROBABILITY ( TIME > ',F10.5,')= ',
     *F10.7)
 1013 FORMAT(10X,'BEGIN TO CALCULATION',/11X,20('-'))
 1990 FORMAT('0',10X,I2,'.',5X,'P(',I2,')=',F10.7)
      END
C
C                                           GET INPUT
C
      SUBROUTINE READ(LP,L,PA,IPB,ICL,IO,IPD,NLAST,T,IPA)
      REAL L(20),PA(20)
      INTEGER IPB(100),ICL(100),IO(100),LP(50),IPD(50),
     *IPA(100)
C                               READ  TIME AND # OF NODE
C
      READ(5,1) T,N
C
C                            READ BACK POINTERLIST,FAILURE
C                                          RATE CODE LIST
C
         READ(5,3)  (IPB(I),ICL(I),I=1,N)
      M=N-1
C
C                           READ ARC ORIGIN LIST AND ARC
C                                PROBABILITY CODE LIST
C
         READ(5,4)  (IO(K),IPA(K),K=1,M)
C
C                          READ # OF PATH AND # OF DIFFERENT
C                            FAILURE RATE AND # OF DIFFERENT
C                                          ARC PROBABILITIES
         READ(5,5)  NLAST,NDIF,MDIF
C
C                               READ LAST POINT LIST (LP) AND
C                                 LAST POINT DEPTH LIST (IPD)
C
         READ(5,6)  (LP(I),IPD(I),I=1,NLAST)
C
C                                      READ FAILURE RATES
C
         READ(5,7)  (L(I),I=1,NDIF)
         READ(5,7)  (PA(I),I=1,MDIF)
      RETURN
    1 FORMAT(5X,F10.3,I5)
    3 FORMAT(5X,10I5)
    4 FORMAT(5X,10I5)
    5 FORMAT(5X,3I5)
    6 FORMAT(5X,10I5)
    7 FORMAT(5X,5F10.5)
      END
C
C        THIS SUBROUTINE COMPUTES NECESSARY LISTS TO
C   CALCULATE THE SURVIVAL FUNCTION OF A CURRENT PATH
C
C
      SUBROUTINE PASS(PA,L,ICL,NPATH,MPATH,IP,LL,NI,IK,P,
     *IPA)
      REAL PA(20),LL(11),L(20)
      INTEGER NPATH(11),MPATH(10),LCODE(10),NI(10),ICL(100),
     *IPA(100)
      P=1.
      DO 1 I=1,IP
```

```
    1 P=P*PA(IPA(MPATH(I)))
          IPP=IP+1
          II=1
          IF(ICL(NPATH(1)).EQ.0)  II=2
          DO 2 I=II,IPP
    2     LCODE(I)=ICL(NPATH(I))
             IK=0
             DO 3 I=II,IPP
             IF(LCODE(I).EQ.0)  GO TO 3
                IK=IK+1
                LL(IK)=L(LCODE(I))
                LLL=1
             JJ=I+1
             IF(JJ.GT.IPP)  GO TO 13
                DO 4 J=JJ,IPP
                IF(LCODE(I).NE.LCODE(J))  GO TO 4
                LLL=LLL+1
                LCODE(J)=0
    4           CONTINUE
   13           NI(IK)=LLL
            LCODE(I)=0
    3       CONTINUE
      RETURN
      END C
C           THIS SUBROUTINE CONTROLS THE
C   COMPUTING OF THE SURVIVAL FUNCTION OF THE CURRENT PATH
C
      SUBROUTINE ONECON(PRO,T,L,NINIT,K)
      REAL A(11,10),L(11)
      INTEGER NI(11),NINIT(11)
      CALL ONEDIS(K,A,NI,L)
      CALL ONEATA(K,A,NI,L,NINIT)
      CALL CALPRO(K,A,NI,L,PRO,T)
      RETURN
      END
C
C                    THIS ROUTINE COMPUTES THE COEFFICIENTS
C         AS EACH IS DISSIMILAR
C
      SUBROUTINE ONEDIS(K,A,NI,L)
      REAL A(11,10),L(11)
      INTEGER NI(20)
      KK=1
    2 A(KK,1)=1.
      NI(KK)=1
      JJ=1
    4 IF(JJ.EQ.KK)  GO TO 6
      A(KK,1)=A(KK,1)*L(JJ)/(L(JJ)-L(KK))
    6 IF(JJ.EQ.K)  GO TO 9
      JJ=JJ+1
      GO TO 4
    9 IF(KK.EQ.K )  GO TO 8
      KK=KK+1
      GO TO 2
    8 RETURN
      END
C
C                               BEGIN TO ADD ONE AT A TIME
C
      SUBROUTINE ONEATA(K,A,NI,L,NINIT)
      REAL L(11),A(11,10)
      INTEGER NINIT(11),NI(11)
      IE=1
   32 IF(NINIT(IE).EQ.NI(IE))  GO TO 99
      NI(IE)=NI(IE)+1
C
C                                       UPDATE  IE
C
```

```
      J=0
 12   NN=NI(IE)-J
      A(IE,NN)=L(IE)*A(IE,NN-1)/FLOAT(NN-1)
      IF(NN.EQ.2) GO TO 20
      J=J+1
      GO TO 12
 20 CONTINUE
         DO 101 I=1,K
         IF(I.EQ.IE) GO TO 101
         A(IE,1)=A(IE,1) +A(I,1)*L(I)/(L(I)-L(IE))
         IF(NI(I).LT.2) GO TO 101
         FACT=1.
         NKK=NI(I)
               DO 102 II=2,NKK
               FACT=FACT*(II-1)
     A(IE,1)=A(IE,1)+A(I,II)*FACT*L(IE)/(L(I)-L(IE))**II
102            CONTINUE
101      CONTINUE
C
C                                            UPDATE  I.NE.IE
C
    I=1
 21 IF(I.EQ.IE) GO TO 26
    A(I,NI(I))=A(I,NI(I))*L(IE)/(L(IE)-L(I))
       IF(NI(I).EQ.1) GO TO 26
         J=1
 24      NNI=NI(I)-J
         A(I,NNI)=(L(IE)*A(I,NNI)-FLOAT(NNI)*A(I,NNI+1))
    *              /(L(IE)-L(I))
         IF(J.EQ.(NI(I)-1)) GO TO 26
         J=J+1
         GO TO 24
 26 IF(I.EQ.K) GO TO 32
    I=I+1
    GO TO 21
 99 CONTINUE
    IF(IE.EQ.K) GO TO 35
    IE=IE+1
    GO TO 32
 35 RETURN
    END
C
C  CALCULATES THE PROBABILITY FOR THE CURRENT PATH
C
    SUBROUTINE CALPRO(K,A,NI,L,PRO,T)
    REAL L(11),A(11,10)
    INTEGER NI(11)
    PRO=0.
    DO 40 I=1,K
    SUM=0.
    NKK=NI(I)
       DO 41 J=1,NKK
       SUM=SUM+A(I,J)*T**(J-1)
 41    CONTINUE
    TTT=-L(I)*T
    PRO=PRO+SUM*EXP(TTT)
 40 CONTINUE
    WRITE(6,1011)
    DO 1000 I=1,K
    NNN=NI(I)
    WRITE(6,1010) L(I),NI(I),(A(I,J),J=1,NNN)
1000 CONTINUE
    RETURN
1010 FORMAT('0',10X,F10.7,5X,I2,5X,2(5(E12.5,2X),/25X))
1011 FORMAT(15X,'LAMDA',7X,'NI',5X,'COEFFICIENTS',/10X,
   135('-'))
    END
```

INPUT DATA FOR THE PROGRAM:

```
T,N        2.          13
NO. 1      0     1      1     1     2     1     3     3     4     1
NO. 6      5     1      6     2     7     3     8     3     9     2
NO. 11    10     3     11     2    12     2
AR. 1      1     1      1     2     2     1     2     2     3     1
AR. 6      3     2      4     3     5     1     5     2     6     1
AR. 11     6     2      7     3
NL,MN      6     3      3
LP1PD      8     3      9     3    10     3    11     3    12     3
LP6PD     13     3
LAMDA      2.0         .5          1.5
PA'S       .25         .75          1.0
```

OUTPUT FROM THE PROGRAM :

```
               BEGIN TO CALCULATION
               --------------------
  1.         P( 1)= 0.0625000
LAMDA    NI  COEFFICIENTS
-----    --- ------------
1.5      2   -0.80000E+02   0.24000E+02
2.0      2    0.81000E+02   0.18000E+02
  2.         P( 2)= 0.0468750
LAMDA    NI  COEFFICIENTS
-----    --- ------------
1.5      2   -0.51200E+03   0.96000E+02
2.0      3    0.51300E+03   0.16200E+03   0.18000E+02
  3.         P( 3)= 0.1406250
LAMDA    NI  COEFFICIENTS
-----    --- ------------
0.5      2   -0.63578E-06   0.11852E+01
2.0      3    0.10000E+01   0.81481E+00   0.22222E+00
  4.         P( 4)= 0.0468750
LAMDA    NI  COEFFICIENTS
-----    --- ------------
1.5      2   -0.51200E+03   0.96000E+02
2.0      3    0.51300E+03   0.16200E+03   0.18000E+02
  5.         P( 5)= 0.1406250
LAMDA    NI  COEFFICIENTS
-----    --- ------------
0.5      2   -0.63578E-06   0.11852E+01
2.0      3    0.10000E+01   0.81481E+00   0.22222E+00
  6.         P( 6)= 0.5625000
LAMDA    NI  COEFFICIENTS
-----    --- ------------
0.5      2   0.59259E+00   0.88889E+00
2.0      2   0.40741E+00   0.22222E+00
   PROBABILITY ( TIME > 2.0 )=  0.0173515
```

APPENDIX D

This section gives the program for the simulation of a system's life mentioned in Section III. The program uses a crude Monte Carlo simulation procedure.

```
C
C       THIS PROGRAM SIMULATES A SYSTEM HAVING
C             THE RELIABILITY FUNCTION:
C
C
C       F(T)=
C     P1   *   F1(T) (=EXP(L1,1)+EXP(L1,2)+.......+EXP(L1,M1))
C   + P2   *   F2(T) (=EXP(L2,1)+EXP(L2,2)+.......+EXP(L2,M2))
C   + P3   *   F3(T) (=EXP(L3,1)+EXP(L3,2)+.......+EXP(L3,M3))
C   +  . . . . . . . . . . . . . . . . . . . . . . . . .
C      . . . . . . . . . . . . . . . . . . . . . . . . . .
C   +  . . . . . . . . . . . . . . . . . . . . . . . . .
C   + PI   *   FI(T) (=EXP(LI,1)+EXP(LI,2)+.......+EXP(LI,MI))
C   +  . . . . . . . . . . . . . . . . . . . . . . . . .
C   +  . . . . . . . . . . . . . . . . . . . . . . . . .
C     . . . . . . . . . . . . . . . . . . . . . . . . . .
C   +  . . . . . . . . . . . . . . . . . . . . . . . . .
C   + PN   *   FN(T) (=EXP(LN,1)+EXP(LN,2)+.......+EXP(LN,MN))
C
C      L( )     : ARRAY FOR ALL LAMDAS.
C      P( )     : ARRAY FOR ALL PROBABILITIES.
C      MI( )    : NUMBER OF EXP IN EACH ROW.
C      MT( )    : FIST NUMBER OF LAMDAS IN THIS ROW.
C      N        : TOTAL NUMBER OF ROWS.
C      PROB     : PROBABILITY OF SYSTEM SURVIVAL AT TIME T.
C      TIME     : TIME
C
C
C
      REAL L(500),P(50),PROB
      INTEGER MI(50),MT(50),MIDO,MIUP,I,J,N
      IX=456378
      CALL READ(L,N,TIME,MI,MT,P)
C
C   CHECK FOR SUM OF P'S EQUAL TO 1.0
C
      TOT=0.
      DO 150 I=1,N
  150 TOT=TOT+P(I)
      IF(ABS(TOT-1.).GT.1.E-5) STOP
      CALL CONTRO(L,N,TIME,MI,MT,P,IX,PROB)
      CALL OUTPUT(L,N,TIME,MI,MT,P,PROB)
      STOP
      END
C
C                                              SUBROUTINE READ
C
      SUBROUTINE READ(L,N,TIME,MI,MT,P)
      REAL L(500),P(50)
      INTEGER MI(50),MT(50)
C
```

```
C  GET THE TIME
C
      READ(5,505) TIME
C
C   GET THE NUMBER OF ROWS
C
      READ(5,501) N
C
C  GET ALL P'S
C
      READ(5,504) (P(I),I=1,N)
C
C  GET ALL LAMDAS IN THE ORDER OF ROW BY ROW
C
      MIDO=1
      MIUP=0
      DO 100 J=1,N
      IF(J.NE.1) MIDO=MIUP+1
      MT(J)=MIDO
      READ(5,520) MI(J)
      MIUP=MIUP+MI(J)
      READ(5,503) (L(I),I=MIDO,MIUP)
  100 CONTINUE
      RETURN
  501 FORMAT(5X,I5)
  503 FORMAT(5F10.5)
  504 FORMAT(5F10.7)
  505 FORMAT(5X,F10.3)
  520 FORMAT(5X,I5)
      END
C
C                                               SUBROUTINE CONTRO
C
      SUBROUTINE CONTRO(L,N,TIME,MI,MT,P,IX,PRO)
      REAL L(500),P(50)
      INTEGER MI(50),MT(50)
      SUM=0.
         DO 1 I=1,N
         NN=1000000*P(I)
         CALL SIMULA(NN,L,MI(I),MT(I),X,IX,TIME)
    1    SUM=SUM+X
      PRO=SUM/1000000.
      RETURN
      END
C
C   SUBROUTINE FOR SIMULATION
C
      SUBROUTINE SIMULA(NN,L,M,MTT,X,IX,TIME)
      REAL L(500),RN(50)
      INTEGER M,I,J,IX
      X=0.
      MMT=MTT+M-1
         DO 11 I=1,NN
         TEST=0.
         CALL LEXPN(IX,RN,M,16807,0)
         JJ=0
               DO 111 J=MTT,MMT
               JJ=JJ+1
  111          TEST=TEST+RN(JJ)/L(J)
   11    IF(TEST.GE.TIME) X=1.0+X
      RETURN
      END
C
C                                               SUBROUTINE OUTPUT
C
      SUBROUTINE OUTPUT(L,N,TIME,MI,MT,P,PRO)
      REAL L(500),P(50)
      INTEGER MI(50),MT(50)
```

```
C
C   PRINT OUT ALL THE SYSTEM
C
        WRITE(6,600)
        MIUP=MI(1)
        WRITE(6,601) P(1), (L(I),I=1,MIUP)
           DO 102 I=2,N
           MIDO=MT(I)
           MIUP=MT(I)+MI(I)-1
           WRITE(6,602) P(I), (L(J),J=MIDO,MIUP)
  102 CONTINUE
        WRITE(6,603) TIME,PRO
        RETURN
C
C   FORMAT STATEMENTS
C
  600 FORMAT('1',5X,'THE SYSTEM IS :')
  601 FORMAT('0',5X,'F(T)=',F10.7,' * ','EXP   L = ',5
     1(F10.4,3X),5(//20X,5(F10.4,3X)))
  602 FORMAT('0',9X,'+ ',F10.7,' * ','EXP   L = ',5(F1
     10.4,3X),5(//20X,5(F10.4,3X)))
  603 FORMAT('0',5X,'PROBABILITY OF SYSTEM SURVIVAL AT
     *T=',F10.5,' IS ',F10.7)
        END
```

INPUT FOR THE PROGRAM :

```
TIME     15.
#ROW      5
  .2          .2          .2          .2          .2
 1.ST     6
  .1          .8          .4          .1          .1
  .8
 2.ND     7
  .2          .2          .5          .6          .5
  .5     .5
 3.RD     9
  .1          .5          .3          .1          .1
  .5          .5          .5          .3
 4.TH     6
  .1          .2          .1          .1          .2
  .2
 5.TH     9
  .1          .2          .4          .1          .2
  .2          .2          .4          .4
```

OUTPUT FROM THE PROGRAM :

```
  THE SYSTEM IS :
F(T)= 0.2 * EXP   L = 0.1    0.8    0.4    0.1    0.1
                      0.8
   +  0.2 * EXP   L = 0.2    0.2    0.5    0.6    0.5
                      0.5    0.5
   +  0.2 * EXP   L = 0.1    0.5    0.3    0.1    0.1
                      0.5    0.5    0.5    0.3
   +  0.2 * EXP   L = 0.1    0.2    0.1    0.1    0.2
                      0.2
   +  0.2 * EXP   L = 0.1    0.2    0.4    0.1    0.2
                      0.2    0.2    0.4    0.4
PROBABILITY OF SYSTEM SURVIVAL AT T=  15.00000 IS   0.911241
```

APPENDIX E

This section reviews some notions which are found in the references for this paper.

E.1 Redundant systems with exponentially lived components:

In reliability analysis, the term system is used to describe a set of components organized to perform some mission. A system is redundant if, in some fashion, some of the components involved act as back up for other components.

A rough definition might be that a system is not redundant if the failure of any one of its components causes the failure of the system, and is redundant if one or more of its components can fail without causing the system to fail. Thus redundant systems have the property that they can suffer damage through the failure of some of their components and still survive. (ESARY[Ref.1])

E.2 First failure in a set of exponentially lived components:

We have n components, each independent from the others, we want to compute the probability that the j.th component fails first.

$$P(\text{j.th component fails first}) = P(T_j < T_i, \forall i \ni i \neq j)$$

$$= \int_0^\infty P(T_i > T_j, \forall i, \ni i \neq j \mid T_j = s)\, \lambda_j e^{-\lambda_j s} ds$$

$$= \int_0^\infty [\prod_{i \neq j} \bar{F}_i(s)\ ] f_j(s)\ ds$$

$$= \int_0^\infty [\prod_{i \neq j} e^{-\lambda_i s}] \lambda_j e^{-\lambda_j s}\ ds$$

$$= \lambda_j / (\sum_{i=1}^n \lambda_i) \int_0^\infty [\sum_{i=1}^n \lambda_i]\, e^{-\sum_{i=1}^n \lambda_i s}\ ds$$

$$= \lambda_j / (\sum_{i=1}^n \lambda_i).$$

E.3 Degeneracy at zero (Zero Distribution):

Let ZERO be the name for the distribution of a random variable that is degenerate at zero.

If $P(T_0 = 0) = 1$, then we say that T_0 has the distribution ZERO, or

$T_0 \sim$ ZERO .

LIST OF REFERENCES

1. Esary, J. D. , Unpublished Course notes and Problem sets for OA 4302, Naval Postgraduate School, Monterey, CA.93940, 1982

2. Repicky, John J., Jr., An Introduction to a Reliability Shorthand, M. S. Thesis, Naval Postgraduate School, Monterey, CA.93940, 1981.

3. Freund, J. E. and Walpole, R. E. , Matematical Statistics, Third edition, 1980.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey California 93940.	2
3.	Departmant Chairman, Code 55 Naval Postgraduate School Monterey California 93940.	1
4.	Professor J. D. Esary, 55Ey Naval Postgraduate School Monterey California 93940.	1
5.	Professor T. Jayachandran, 53Jy Naval Postgraduate School Monterey California 93940.	1
6.	Lt. J. G. Sadan Gursel Inonu Cad. No. 10 Kesan - Edirne / TURKEY	1
7.	Ortadogu Teknik Universitesi Yon Eylem Arastirmasi Bolumu Ankara/TURKEY	1
8.	Dz. K. K. ligi Bakanliklar, Ankara / TURKEY	2
9.	Lt. Deniz CORA Oyak Sitesi 27. Blok Da. 1 Yeni Levent, Istanbul / TURKEY	1
10.	Deniz Harp Okulu K. ligi Heybeliada, Istanbul / TURKEY	1

Thesis 198314
G868 Gursel
c.1 Some computer algorithms to implement a reliability shorthand.